

PARALLELIZATION OF ROCKET ENGINE SIMULATOR SOFTWARE

(P.R.E.S.S.)

FINAL REPORT

Principal Investigator

Dr. Ruknet Cezzar, Associate Professor
Department of Computer Science
Hampton University
Hampton, Virginia 23668
email: cezzar@cs.hamptonu.edu

Technical Officer

Dr. Donald Noga
Mail Stop 501-2
NASA Glenn Research Center
21000 Brook Park Road
Cleveland, Ohio 44135

Grant Number: NAG3-1792

Grant Period: 3 Years

Start date: October 19, 1995
End Date: October 18, 1998
Extension Date: October 18, 1999

(October 31, 1999)

TABLE OF CONTENTS

<u>Section</u>	<u>Subject</u>	<u>Page</u>
1.	Background	1
2.	Research Progress Overview	1
3.	Project History, Phases, and Activities	2
	3.1 Phase I (October 19, 1995 - October 18, 1996)	
	3.2 Phase II (October 19, 1996 - October 18, 1997)	
	3.3 Phase III (October 19, 1997 - October 18, 1998)	
	3.4 No-Cost Extension Period (Oct 19, 1998 - Oct 18, 1999)	
4.	Proposed Work for the Future	9
5.	Conclusion and Recommendations	9
	REFERENCES AND BIBLIOGRAPHY	11

1. Background

Parallelization of Rocket Engine System Software (PRESS) project is part of a collaborative effort with Southern University at Baton Rouge (SUBR), University of West Florida (UWF), and Jackson State University (JSU). The project has started on October 19, 1995, and after a three-year period corresponding to project phases and fiscal-year fundings by NASA Lewis Research Center (now Glenn Research Center), has ended on October 18, 1998.

The one-year no-cost extension period was granted on June 7, 1998, until October 19, 1999. The aim of this one year no-cost extension period was to carry out further research to complete the work and lay the groundwork for subsequent research in the area of aerospace engine design optimization software tools. The previous progress for the research has been reported in great detail in respective interim and final research progress reports, seven of them, in all. While the purpose of this report is to be a final summary and an evaluative view of the entire work since the first year funding, the following is a quick recap of the most important sections of the interim report dated April 30, 1999:

2. Research Progress Overview

While the aim of this report is to give an overall view of the entire four-year period of the research activity involving PRESS, including the one-year no-cost extension period, the following is a quick summary of the research progress since the filing of the last report, namely the aforementioned interim report dated April 30, 1999:

- Search for design optimization tools for aerospace engines
- Contact with the Engineous Software, Inc. for evaluation of iSIGHT 4.02 software
- Obtaining iSIGHT from Engineous Software, Inc. after they established a new policy for sale of iSIGHT to educational institutions such as Hampton University (Hampton University was the first in this regard)
- Making the license and software agreement for the use of iSIGHT between Hampton University and Engineous Software, Inc. for a period of one year.
- The attendance of the principal investigator in a five-day iSIGHT training workshop held at the home office of the Engineous Software, Inc. in Morristown, North Carolina (July 12-17, 1999)
- Development of submission of AeroSpace Engine Numeric Design Optimization and Synthesis (ASPENDOS) proposal (October 1, 1999)

3. Project History, Phases, and Activities

In this section we would like to provide a general idea of the different phases of the project corresponding to one-year funding periods and the one year no-cost extension period. The description will give an idea of how the project evolved over the four year period since its first funding back in October 19, 1995.

3.1 Phase I (October 19, 1995 - October 18, 1996)

As is articulated in the related interim report [1] and research summary report [2] at the end of this period, the first year activities primarily focused on parallelization of the Fortran based software for the Two-Dimensional Kinetics (TDK) package. The interim report also mentions the activities related to gearing up for the project by purchasing necessary computer equipment and staffing. Toward this end, after the assessment of the then existing computer hardware and software at Hampton University, the purchase of PC hardware and software such as Borland's C++ were discussed. Moreover, stipend support was provided for one graduate student, Mr. Ilesanmi Omawole, for Summer 1996 with an optional continuation of support during the academic year 1996-97. Mr. Ilesanmi Omawole was then a transferee to the computer science graduate program from the Department of Electrical Engineering, and had engineering background and proficiency with Fortran programming.

The following generic goals were established and partially realized during the second half of this phase:

- Common platform for diverse software

Translations (source code), porting, reconfiguration on Hampton University's computing facilities

- Improve performance and code efficiency

Parallelizations, vectorizations, executable code optimizations on Hampton University's supercomputer Cray X-MP and parallel computer NCUBE-64

- Demonstration and Promotion of Rocket Engine Software Suite

GUI Improvements using visual and animation languages. Remote access and execution facilities on the Internet. User guides and documentation on the World Wide Web home pages

During the second half of this phase, the TDK package was checked out as regards its size, complexity, and suitability for parallelization. TDK was found to be old legacy software with large size consisting of 280 Fortran modules and great complexity due to dated and unreliable documentation developed over its history since 1978. This issue has been discussed in some detail in the year-end progress report. Moreover, there has been an effort toward translation of the Fortran based TDK software into C++, an object-oriented language. To this end, about one-third of the code has been translated into C and C++. In this effort, we enlisted the help of another graduate student, Ms. Chenhong Lu. The goal was to translate the entire TDK software which was Fortran based and on Unix platform into Borland C++ on PC platform. As well articulated with excerpts of email exchanges in the year-end progress report, this goal could not be accomplished because of the unwieldy size and complexity of the TDK package and incomplete and conflicting documentation which was dated and almost obsolete by today's standards.

Through email exchanges with Ken Davidian who was the coordinator for the project, it was decided that the project be lent another smaller size software with the requisite software agreement. To that end, it was decided that the Turbine Design and Pump Design packages, TURBDES/PUMPDES, along with the shared module GASP (Gas Properties) be lent to Hampton University under another software agreement for the purpose of experimentation toward fast parallel execution, translation into object-oriented languages, and support for collaborative computing in distributed fashion. An important objective and recommendation for the second phase, at that time, was to make the Fortran based legacy software, be it TDK or TURBDES/PUMPDES, a user-friendly one with GUI interfaces, and at the same time, a well-described and documented software through translation into an object-oriented language such as C++.

3.2 Phase II (October 19, 1996 - October 18, 1997)

During the first half of this phase, while efforts to obtain and install a working version of the PUMPDES/TURBDES/GASP package was in progress, the efforts to streamline and clarify TDK software continued. In this regard, the following is a verbatim excerpt found at the introductory section of the interim report dated April 28, 1997 [3]:

Short of a coordinated effort involving various NASA centers and organizations within the Lewis Research Center and possibly other universities involved in the RENS collaborative research effort, the report outlined the plans for the second year funding for reworking of the current Fortran based TDK software. In a nutshell, the effort for the second year funding was characterized as: *major reworking, streamlining, redesign, and re-documentation of the TDK software, and the development of entirely new documentation*. Furthermore, the limitations of PC platform which was necessary for GUI interfaces were mentioned.

As part of staffing for this phase, Ms. Chenhong Lu, who is a Chinese national with F-1 student visa status, joined the graduate program in Fall of 1996 was hired as research assistant. She had a Master's degree in Mathematics and excellent background in computer sciences. Due to a family emergency in her native country, however, she could only work in Spring and Summer semesters of 1997. The principal investigator during Fall 1996 and Spring 1997 semesters and Ms. Chenhong Lu during Spring 1997 have worked exclusively on the reworking of the TURBDES/PUMPDES rocket engine design simulation software cluster. This research effort during the first half of this phase, from October 19, 1996 to April 28, 1997 focused on downloading of the TURBDES/PUMPDES/GASP software cluster from Lewis Research Center and reconfiguration on HU's Cray Y-MP and SunOS NFS systems. While both the PI and the research assistant were getting familiar with the physical and mechanical design of liquid-propelled rocket engines through references [8, 9]. The FORTRAN code which invoked approximately 15 modules was checked, compiled, debugged to run on both on Sun spark based Unix platforms as well as Windows based PC platforms. Most of the effort, however, was expended on understanding the documentation for these packages and verification of the information contained therein. Although the documentation was clearer and concise as opposed to TDK's, there were still some ambiguities and discrepancies.

At the end of the first half of this phase, a significant activity was the attendance at R.E.N.S. Meeting and Review of Research Progress, University of West Florida, Pensacola, Florida, held during January 16-17, 1997. The meeting and research progress seminars and presentations involved all the participants of the overall RENS research effort. There, the Principal Investigator presented the goals and plans for Parallelization of Rocket Engine System Software. The following is a verbatim recap of parts of that presentation:

PRESS Project-Goals and Strategies (Second Year)

- Common platform for diverse software
Porting and reconfiguration of RENS related software on HU computing facilities
- Accomplished for TDK software on Sun Workstations and Cray Y-MP (in FORTRAN)
- For TURBDES/PUMPDES software, glitches involving plotting subsystem require further testing and debugging work
- Improve performance and code efficiency through vectorization, optimization, and parallelization on HU's super/parallel computers
- Strategy:
 - Focus on TURBDES/PUMPDES with smaller scope (5 modules each as opposed to 282 for TDK software) and better documentation
 - Accomplish vectorizations, optimizations, and parallelization using CRAY Y-MP cf77 Fortran compiler with impressive benchmark results.

At the RENS Meeting, however, the new impetus for the RENS projects in general, and PRESS in particular, has shifted in two important ways. One was closer alignment with the work on Numerical Propulsion System Simulator (NPSS) through cooperation and collaboration with LeRC ACLU organization. The other was to see whether and how NASA's various rocket design software can be run over local and intra nets without any radical efforts for redesign and translation into object-oriented source code. There were also suggestions that the Fortran based code be encapsulated in C++ code thereby facilitating reuse without undue development effort. The details are covered in the aforementioned section of the interim report filed on April 28, 1997.

During the second half of the phase II, while, Ms. Chenhong Lu, graduate research assistant for this project which we refer to as the PRESS project, continued to concentrate her efforts on the previously obtained TURBDES/PUMPDES rocket engine design packages (9), the principal investigator has worked on establishing a foundation for future work toward distributed access and use of NASA's rocket engine design software packages. The efforts, at the time, was categorized as:

1. Parallel and distributed computing tools and packages for sharing of rocket engine software across local area and corporate intranets
2. Investigation of C++ wrappers around Fortran code and related issues of mixed-language programming interfaces involving Fortran, C, and C++
3. Checkouts and correct installation of Rocket Engine Transient Simulator (ROCETS) on Unix platform to enhance understanding of rocket components and design issues

where further details can be found in the associated research progress report [4].

3.3 Phase III (October 19, 1997 - October 18, 1998)

The most significant part of Phase III was the new directions which resulted from the RENS Meeting in Pensacola, Florida in January 1997. With Mr. Joseph A. Hemminger taking over the coordination of the Rocket Engine Numeric Simulator project, the new research impetus was on providing parallel and distributed computing environment for rocket engine design and simulation software. Moreover, there was to be placed more emphasis on collaborative computing regardless of the high level languages in which the legacy software was developed. The earlier emphasis, during the first and second project phases corresponding to the two fiscal year funding period, had primarily been on reworking of the existing Fortran based software for better performance and user interfaces and on attempts to translate them into a more modern object oriented language C++.

As is detailed in the interim report for the first half of this period, the principal investigator primarily worked on acquisitions of public-domain distributed computing packages MPI and PVM for installation on the Hampton University's LAN. Furthermore, since the shared GASP (GAS Properties) module was found to be inadequate for running the TURBDES/PUMPDES simulator software, preparations were made for the acquisition of a new version referred to as GASPSHEER at the time because of the work done by Dean Sheer of NASA Lewis Research Center.

During that period, work on wrapping of Fortran code by C++ commenced. For that work, on the PC platform, Microsoft Fortran 5.1 and Borland C++ 5.0 were utilized. On the SunOS-Sparcstation based Unix platform, GNU g++ compiler was used for C++ and standard f77 was used for Fortran 77. Numerous issues came up because of the development of much of the legacy software was developed in VAX VMS Fortran on mainframe platforms and Lahey Fortran on PC platforms.

One activity during that period was to provide the principal investigator access to the ACCL's cluster of computers at NASA Lewis Research Center. The purpose was to be able to run working versions of the distributed computing packages MPI and PVM on both HU's local area network as well as at NASA LeRC's ACCL LACE cluster. Some of the challenges regarding remote access and execution to the LACE cluster had been discussed in detail in the respective interim report [5]. The key part of the interim report dealt with the acquisition of the working version of GASP (gas properties) module, referred to as GASPSheer. That was necessary for debugging and obtaining an executable for PUMPDES/TURBDES software which has been lent earlier to Hampton University, under a software agreement. There is also the discussion of PI's presentation at the Fifth Annual HBCU Conference, April 9-10, 1997, Cleveland, Ohio. The presentation basically outlines distributed computing strategies for NASA's Fortran-based software, in particular, the RENS and NPSS software at Lewis Research Center. The presentation focuses on the major challenge involving lack of support for Fortran-based software by distributed computing standards and software tools such as CORBA (Common Object Request Broker Architecture). As its name indicates, this most popular distributing computing standard, only supports object-oriented languages such as C++, Smalltalk, and Java. More recently, it also provides limited support for Ada. Finally, there was some discussion of the work on C++ wrappers, in our case, using Microsoft Fortran 5.1 and Borland C++ 5.0, on PC platform.

During the second half of the Phase III, the main work involved experimentation with MPI, and both non-GUI and GUI versions of PVM (Parallel Virtual Machine). Those were, at the time, the most popular public-domain distributed computing tools. Both MPI and PVM were, and still are, geared towards Unix based local area networks, and aim to provide parallel execution over a local net. Once that is accomplished, the next step is provide the same capabilities in an intra-net environment where there may be one or more heterogenous local area networks. In that case, there is a need to use application tools which are based on CORBA (Common Object Request Broker Architecture) standard.

The work had a specific purpose of demonstrating the use of MPI using a working version of a rocket engine simulation software. Initially, the aim was to use a working version of

PUMPDES or TURBDES software using the new GASP obtained from NASA LeRC with a new software agreement. Later, with acquisition of what became known as SOURCCDS-folder modules under another software agreement, it became clear that using the main executable module of that package would be more productive. That main module is RESSAP.FOR, where RESSAP is an acronym for Rocket Engine System Sizing and Analysis and Program. RESSAP.FOR is a preliminary source code, and has not been thoroughly tested. However, RESSAP calls both PUMPDES and TURBDES modules, as well as other modules. Thus, it appeared that RESSAP would be a good candidate for experimentation with MPI.

The year-end research progress report [6] provided a detailed description of how MPI can be used to run RESSAP code in parallel over a local net. In the description, the demo implemented at HU's LAN was used as an example. Unfortunately, this demonstration package was not readied in time for presentation at the Fourth Annual HBCU conference held during April 8-9, 1997 in Cleveland, Ohio. Instead, the postal and oral presentations focused on the then distributed computing packages MPI (Message Passing Interface), PVM (Parallel Virtual Machine) [10,11], and CORBA (Common Object Request Broker Architecture) [12-17]. As was emphasized during that presentation, CORBA is a rapidly evolving standard for corporate intranets and should be considered in any distributed computing effort. Indeed, one key aspect of our presentation at the conference was pointing out the lack of support for Fortran based software in CORBA standards. This primarily stems from the fact that Fortran is not an object-oriented language. We have also pointed out various alternatives such as C++ wrappers around Fortran code so that tools like CORBA can be utilized.

3.4 No-Cost Extension Period (Oct 19, 1998 - Oct 18, 1999)

During the one-year no-cost extension period, while still working on the intra-net middle-level distributed computing involving rocket engine design simulator software using the Common Object Request Broker Architecture (CORBA) standards, we also wanted to acquire the I-Sight software for a licensing period of one year which would roughly correspond to the extension period from October 19, 1998 to October 18, 1999. Therefore, the two main activities have been the efforts toward acquisition of the iSIGHT software and the participation in the HBCU conference for presenting the emerging trends in distributed computing and for exploring fruitful areas of research after the PRESS project extension period ends in October 18, 1999.

Through contacts with Engineous marketing representative Mark A. Ondracek, referred to us by Joe Hemminger, we explored the possibility of purchase of iSIGHT for the PRESS project. At first, the prices in the range of \$ 12,000 - \$ 14,000 was too high for the PRESS budget. We then inquired whether the Engineous, Inc. had any educational discounts. At the time, there was no such program. After several weeks of discussions and a decision by the management of Engineous, Inc., we were offered the purchase of the software at a substantially reduced price. Accordingly, based on a related quotation from Engineous, Inc., we have issued a requisition in the amount of \$ 5,150.00 to purchase the software for the licensing period of one year. Subsequently, the Engineous, Inc. requested to sign a software agreement with Hampton University regarding the licensing and use of the software. This agreement was concluded on February 25, 1999.

Subsequently, the principal investigator attended a five-day workshop as part of the aforementioned iSIGHT license. The workshop was held in the home offices of Engineous Software, Inc. in Morristown, North Carolina. The period of attendance was from July 12 through July 17.

The first two days involved the introduction to iSIGHT application development and was taught by Randolph Elliott. It involved seven laboratory exercises on various features of iSIGHT such as task management, input parsing, output graphs and charts, optimization methods, and rules engine. Typical simulation codes for engineering designs from i-beams to electrical-circuit balancing were used as examples.

The last three days, conducted by Gene Farelly who was also helpful in the installment of the iSIGHT package on Hampton University's NT-based servers, involved advanced topics such as verify-parse, approximations, design of experiments, and Multi-Disciplinary Optimization Language (MDOL). This is a proprietary language especially developed for iSIGHT for creating description files which are interpreted by the iSIGHT task manager. Finally, during this session, tcl (tool command language) and perl scripting language were briefly discussed. The latter are Unix based tools for more flexible application implementations for iSIGHT. In a nutshell, iSIGHT is a tool for the design engineer for balancing and optimizing his/her design involving multiple simulation codes. Presently in its version 4.02, iSIGHT can run on both Unix and NT platforms, and provide impressive flexibility and efficiency in terms of saving time and effort in engineering design. In particular, in the case of our research efforts, iSIGHT is a most suitable and valuable, perhaps indispensable, tool in running multiple simulation codes involving aerospace engine analysis and design. The interim report for the extension period [7] provides further details about iSIGHT.

On the second half of the extension period, the principal investigator has recently attended the Historically Black Colleges and Universities/Other Minority Institutions Research Conference sponsored by the Ohio Aerospace Institution (OAI). The conference took place on April 14-15 and the PI was the first presenter on April 14. The topic of our presentation was the recent emerging trends as regards CORBA. There has been rapid progress in the evolution of the CORBA with respect to standards, revisions and new standards, and most important, there has been many viable applications and vendor offerings utilizing CORBA standards to provide distributed enterprise solutions over heterogeneous networks. As was articulated in the presentation, Common Object Request Broker Architecture (CORBA) is becoming the most popular tool for enterprise solutions involving corporate intranets and the Internet. There has been many vendor offerings providing interoperability of software written in different high-level languages over heterogeneous networks. Although Fortran is not included in the Interface Definition Language (IDL) of CORBA, there is still a potential for Fortran based software to be integrated in as a distributed computing software using the existing and planned CORBA utilities and facilities. This, in turn, requires some kind of wrapping of Fortran code within an object oriented language such as C++ or Java. Whether or not the rocket engine design software will take such an approach is, at this time, an open question.

4. Proposed Work for the Future

After the acquisition of iSIGHT software mid-way through the no-cost extension period, and having undergone an intensive training in its use and intricacies, it was felt that this tool be thoroughly evaluated and investigated for applications involving rocket and aerospace engine design simulation codes. Since there has been numerous legacy codes in the area of rocket and aerospace engines, of which four has been lent Hampton University under respective software agreements, iSIGHT can provide a good framework for efficient use of all of those simulation codes. Moreover, in engineering optimizations involving the design of aerospace engines and engine components such as turbines, pumps, nozzles, and the like, a tool such as iSIGHT can be a great time saver. Finally, iSIGHT can foster collaborative computing especially as regards to its library functions which saves all historical data involving simulation runs and provides them back on demand.

With these considerations, the principal investigator developed a three-year proposal and submitted it to perusal of the technical staff at NASA Glenn Research Center on October 1, 1999. The proposal envisions innovative uses of iSIGHT, including some extension to its rule-based expert-systems subcomponent. Furthermore, the proposal envisions iSIGHT being a main reference point or framework in more coherent use of all of the existing legacy simulation codes developed as part of the RENS project. This more coherent use mentioned naturally enhances collaborative computing since, through iSIGHT library facilities, sharing of information involving various simulation runs of clusters of Fortran based modules is facilitated. Finally, as is being worked for the future iSIGHT versions, the proposal also covers the issues involving the CORBA compliancy. For many applications which support distributed computing, such as the widely encountered client/server models on local networks, CORBA compliancy is becoming a hot issue. The reason for this is clear: CORBA compliant applications can easily be extended to intra-nets which are, according to the latest trends, becoming very important.

5. Conclusions and Recommendations

The PRESS project has contributed in areas of parallel and distributed computing involving rocket engine simulation codes. Many aspects of this has been covered in great detail in the research progress reports submitted. Among those are the use of MPI for running the simulation code over a local area network and the possibilities for running the simulation codes over intermediate level intranets through CORBA compliant tools such as Iona's Orbix and Visigenic's Visibroker. Moreover, attempts were made with some progress, albeit limited, in wrapping Fortran based code in C++ so as to provide an object-oriented design environment for the software. Such an approach makes it possible to use facilities such as CORBA, but at a minimum, makes it possible to have better graphical user interfaces (GUIs). We have also attempted to clarify and streamline some of the packages lent to Hampton University under software agreements. However, progress in this area has been hampered by the nature of documentation which tended to be old fashioned and hard to understand.

In the later periods, we have also seriously investigated ways of collaborative computing involving rocket and aerospace engine design simulation codes, and to this end, we have acquired the iSIGHT software as the main research tool for optimizations.

As regards recommendations, the last viewgraph of our presentation at the Fourth Annual HBCU Conference (April 8-9, 1997) is worth reiterating, nothing that not much has changed in trends with respect to distributed computing.

Why not a more radical approach?

- Redesign software based on Fortran 77 and Older along object-oriented principles
- Translate the source code to an object-oriented programming language (C++ or Java)

Biting the Bullet: Cons and Pros

Cons:

- Far greater price to pay for "going distributed"
- Performance (Fortran code probably runs faster)
- Physics and engineering-oriented NASA staff's close familiarity with Fortran is foregone

Pros:

- Redesign and reworking will improve the quality
- Much easier to provide graphics-based user interfaces
- Going distributed is much easier
- Clear goals for and better utilization of grantees..

Moreover, only when NASA decides to convert at least some of the legacy code in the area of rocket and aerospace engine design into an object-oriented language, tools such as CORBA which are rapidly evolving and growing in popularity can be utilized. Moreover, CORBA is a viable distributed computing and application integration tool and would enhance the collaborative computing as regards aerospace engine simulation codes. As for MPI and PVM which is not further discussed in this report, their scope is narrow, and covers only the parallel machines or local networks with distributed operating systems. By the distributed operating system, we mean something like Sun's NFS(Network File System, Solaris threading facilities, and the like.

REFERENCES AND BIBLIOGRAPHY

- [1] **Interim Report** submitted to NASA Lewis Research Center, Technical Officer: Donald Noga, Principal Investigator: Dr. Ruknet Cezzar (April 14, 1996).
- [2] **Research Summary Report** submitted to NASA Lewis Research Center, Technical Officer: Donald Noga, Principal Investigator: Dr. Ruknet Cezzar (October 4, 1996).
- [3] **Interim Report** submitted to NASA Lewis Research Center, Technical Officer: Donald Noga, Principal Investigator: Dr. Ruknet Cezzar (April 28, 1997).
- [4] **Research Summary Report** submitted to NASA Lewis Research Center, Technical Officer: Donald Noga, Principal Investigator: Dr. Ruknet Cezzar (September 2, 1997).
- [5] **Interim Report** submitted to NASA Lewis Research Center, Technical Officer: Donald Noga, Principal Investigator: Dr. Ruknet Cezzar (April 21, 1998).
- [6] **Research Summary Report** submitted to NASA Lewis Research Center, Technical Officer: Donald Noga, Principal Investigator: Dr. Ruknet Cezzar (October 18, 1998).
- [7] **Interim Report** submitted to NASA Lewis Research Center, Technical Officer: Donald Noga, Principal Investigator: Dr. Ruknet Cezzar (April 30, 1999).
- [8] Huzel, D. K. and Huang, D. H. *Modern Engineering for Design of Liquid-Propellant Rocket Engines (Second Printing)*, AIAA, Inc., Washington, D.C. 20024, (1992).
- [9] Sutton, G. P. *Rocket Propulsion Elements: An Introduction to Engineering of Rockets (6th Edition)*, John Wiley and Sons, Inc., New York, (1992).
- [10] Snir, M., Otto, S.W., Huss-Lederman, S., Walker, D.W., Dongarra, J. *MPI: The Complete Reference*, The MIT Press, Cambridge, Massachusetts (2nd Printing, 1997).
- [11] Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R., Sunderam, V. *PVM: Parallel Virtual Machine, A User's Guide and Tutorial for Networked Parallel Computing*, The MIT Press, Cambridge, Massachusetts (Third Printing, 1996).
- [12] Guttman, M. and Applebaum, R. "The Next Generation of CORBA," *Component Strategies*, August 1998, pp. 40-46.
- [13] Seetharaman, K. "The CORBA Connection," *Comm. of ACM*, vol. 41, no. 10 (October 1998)
- [14] Siegel, J. "CORBA and the OMA in Enterprise Computing," *Ibid.*
- [15] Vinoski, S. "New Features for CORBA 3.0," *Ibid.*

- [16] Schmidt, D. C. "Evaluating Architectures for Multithreaded Object Request Brokers," Ibid.
- [17] Haggerty, P. and Seetharaman, K. "The Benefits of CORBA-based Network Management," Ibid.
- [5] Williams, A., Follen, G., Claus, R., Blech, R, et. al. "Key Technologies for Implementing An NPSS," Technical Memorandum, NASA Lewis Research Center, February 1997, [DRAFT].
- [12] Cezzar, R. *A Guide to Programming Languages: An Overview and Comparison*, Artech House Publishers, Inc., Norwood Mass. 02062, 1995 (Chapter 15: Mixed-Language Programming).
- [13] Hemminger, J. A. Space Transportation Propulsion Integrated Design Environment (STP/IDE), presentation at NASA Lewis Research Center, July 29, 1998.
- [14] Follen, G. Williams, A., Blech, R, and Drei, D.V. (NASA Lewis), Apel, A. (P&W East Hartford), Byrd, R. (P&W, West Palm Beach), Gardocki, M. (G.E. Aircraft Engines), Crawford, N. (AlliedSignal Engines), Ashleman, R. (Boeing), McNelly, M. (Allison Engine Co.) "Numerical Propulsion System Simulation Architecture Definition," NASA TM 107343, November 1996.
- [15] Miller, B., Szuch, J. R., Gauigier, R. E., Wood, J.R. "A Perspective on Future Directions in Aerospace Propulsion System Simulation," Lewis Research Center, NASA TM 102038, 1989.
- [16] Reese, D. S., and Luke, E., "Object Oriented Fortran for Development of Portable Parallel Programs, Mississippi State University, NASA Grant NAG3-1073 and NSF cooperative agreement ECD-8907070).
- [17] Blech, R.A. and Arpasi, D.J., "An Approach to Real-Time Simulation Using Parallel Processing," , NASA TM 81731, 1981.
- [18] Jell, Thomas, "Building Multimedia Application Systems Using Component Technology," WEB APPS Solutions Report W4, Object Expo/JAVA Expo '97, New York City, New York, May 2, 1997 (presentation viewgraphs).